Title: SMOS L2 OS OTT Post-Processor Software User Manual

Doc code: SO-MA-ARG-GS-0081

Issue: 0

Revision: 2

Date: 29 November 2013

	Name	Function	Company	Signature	Date
Prepared	SMOS Team		Expert Support Laboratories		
Reviewed	P. Spurgeon	Project manager	ARGANS		
Released	S. Delwart	Project manager	ESA		

All rights reserved ARGANS © 2013

Change Record

Issue	Revision	Date	Description	Approval
0	1	18-06-2013	First version	

Issue	Revision	Date	Description	
0	2	29-06-2013	Minor corrections from JCD comments 18-07-2013	
0	2	13-09-2013	Updated return codes	
0	2	11-10-2013	Added to section 2.1 a description of AUX_DTBCUR initialisation and updating; corrected job order filename in section 5.3	
0	2	29-11-2013	build v611	
0	2	29-11-2013	corrected doc version number in header & first page	

All rights reserved ARGANS © 2013

Table of Content

1. INTRODUCTION 4

1.1 GENERAL 4
1.2 GUIDE TO THIS DOCUMENT 4
1.3 REFERENCE DOCUMENTS 4
1.4 APPLICABLE DOCUMENTS 5
1.5 ABBREVIATIONS 6

2 OVERVIEW 7

2.1 INTRODUCTION 7

3 INSTALLATION 8

3.1 INSTALLATION OF THE OTT POST-PROCESSOR 8
3.2 TREE ORGANIZATION 8

4 L2 OS POST-PROCESSOR DEPENDENCY MODEL 10

4.1 L2 OS POST PROCESSOR AUXILIARY DATA 10
4.2 CONVENTIONS 11
4.3 OTT POST PROCESSOR PROCESSING 11

5 OPERATIONAL CONFIGURATIONS 13

5.1 INTRODUCTION 13
5.2 THE TASK TABLE 13
5.3 JOB ORDERS 13
5.4 THE PRIVATE CONFIGURATION FILE 13
5.5 THE MAIN CONFIGURATION FILE (AUX_CNFOSF) 14

6 RETURN CODES 15

7 GSL LOG INFO MESSAGES 20

All rights reserved ARGANS © 2013

1. Introduction

1.1 General

This document is the user manual of the SMOS L2 OS post-processor. The post-processor has been developed to be operated on Linux 64 bit platforms. It is operated from the command line interface, provided the configuration of the processor is valid. Configuration of the software includes many files, switches, selectors and parameters that are delivered in the distribution packages.

1.2 Guide to this document

This document addresses the L2 OS post-processor installation and use.
For detailed information on the processor, numerous documents are released along with the software:

- For scientific aspects of the SMOS L2 OS processing, see the Algorithm Theoretical Baseline Document [R.D. 1].
- For inputs and output data, including the configuration files, see the L2 products specification document [R.D. 4].
- For details on the data processing, see the Detailed Processing Model [R.D. 3].
- For details on the material released along with the processor, see the Operational Processor Software Release Document SO-RN-ARG-GS-0019.

1.3 Reference Documents

Reference	Title	Code
[R.D. 1]	SMOS L2 OS Algorithm Theoretical Baseline Document	SO-TN-ARG-GS-0007
[R.D. 2]	SMOS L2 OS Operational Processor Computation Resources Requirements	SO-TN-ARG-GS-0011
[R.D. 3]	SMOS L2 OS Detailed Processing Model	SO-TN-ARG-GS-0008
[R.D. 4]	SMOS Level 2 and Auxiliary Data Product Specification	SO-TN-IDR-GS-0006
[R.D. 5]	SMOS L2 OS Architectural Design Document	SO-DD-ARG-GS-0017
[R.D. 6]	SMOS L2 OS Detailed Design Document	SO-DD-ARG-GS-0012 [1]
[R.D. 7]	SMOS L2 OS Software Verification and Validation Plan - Unit Test	SO-TP-ARG-GS-0013
[R.D. 8]	SMOS L2 OS Table Generation Requirement Document	SO-TN-ARG-GS-0014
[R.D. 9]	SMOS L2 OS Prototype Processor Software User Manual	SO-MA-ARG-GS-0021
[R.D. 10]	SMOS L2 OS Prototype Processor Software Release Document	SO-RN-ARG-GS-0022
[R.D. 11]	SMOS Level 1 and Auxiliary Data Product Specification	SO-TN-IDR-GS-0005

[1] Detailed Design Document is available in electronic version only. The DDD is a web site dedicated to the documentation of the Operational Processor.

All rights reserved ARGANS © 2013

Reference	Title	Code
[R.D. 12]	XML Read/Write API Software User Manual	SO-ID-IDR-GS-0009
[R.D. 13]	General Software Library User Manual	SO-MA-IDR-GS-1002
[R.D. 14]	SMOS L2 OS Input / Output Data Definition	SO-TN-ARG-GS-0009
[R.D. 15]	SMOS L2 OS Architectural Design Document	Same as [R.D. 6]
[R.D. 16]	SMOS L2 OS Algorithm Validation Plan	SO-TP-ARG-GS-0015
[R.D. 17]	SMOS L2 OS Algorithm Validation Test Procedure Report	SO-TR-ARG-GS-0016
[R.D. 18]	SMOS L2 OS Parameter Data List	SO-TN-ARG-GS-0010
[R.D. 19]	Earth Explorer Mission CFI Software Explorer_Lib Software User Manual	EE-MA-DMS-GS-0003
[R.D. 20]	SMOS L2 OS Operational Processor Software User Manual	SO-MA-ARG-GS-0018
[R.D. 21]	SMOS L2 OS Prototype Processor Software Release Document	Same as [R.D. 10]
[R.D. 22]	Earth Explorer Mission CFI Software Mission Conventions document	CS-MA-DMS-GS-0001
[R.D. 25]	Software Verification and Validation Plan - Acceptance Test	SO-TP-ARG-GS-0025
[R.D. 26]	Software Verification and Validation Plan - System Test	SO-L2-SSS-ACR-0009
[R.D. 27]	SMOS L2 OS Operational Processor Auxiliary Data Files Configuration Control	SO-CS-ARG-GS-0039

A complete list of applicable documents with versions & dates is available in the Operational Processor Software Release Document SO-RN-ARG-GS-0019.

1.4 Applicable Documents

Reference	Title	Code
[A.D. 1]	SMOS L2-OP Operational Constraints	SO-TN-GMV-GS-4402
[A.D. 2]	PDPC CORE Generic IPF Interface Control Document	SO-ID-IDR-GS-1001
[A.D. 3]	Earth Explorer File Format Standards	PE-TN-ESA-GS-0001
[A.D. 4]	Level 2 Processor ICD and Operational Constraints	SO-ID-IDR-GS-0003
[A.D. 5]	Space Engineering, Software – Part 1: Principles and requirements	ECSS-E-40 Part 1b
[A.D. 6]	Space Engineering, Software – Part 2: Documents Requirements Definition	ECSS-E-40 Part 2b
[A.D. 9]	DPGS Master ICD	SO-ID-IDR-GS-0016
[A.D. 10]	XML schema guidelines	SO-MA-IDR-GS-0004
[A.D. 11]	DPGS schema versioning	SO-TN-IDR-GS-0024
[A.D. 12]	SMOS DPGS acronyms	SO-TN-IDR-GS-0010
[A.D. 13]	SMOS L2 Open Prototype Requirements and Architecture	SO-RS-GMV-GS-4401

All rights reserved ARGANS © 2013

A complete list of reference documents with versions & dates is available in the Operational Processor Software Release Document SO-RN-ARG-GS-0019.

1.5 Abbreviations

The list of acronyms is available in the Operational Processor Software Release Document SO-RN-ARG-GS-0019.

All rights reserved ARGANS © 2013

2 Overview

2.1 Introduction

Figure 1 shows the organisation and data flow of the L2OS processor and OTT post-processor. The L2OS post-processor architecture is similar to that of the L2OS processor (see OPSUM, R.D.20). The post-processor is executed from the command line using a job order specifying input files and output directories.

The L2OS processor ingests L1c products and a number of other files including OTTs (AUX_OTTxF/D). These OTTs are updated frequently (nominally every day) by the OTT post-processor. The OTT post-processor ingests all the AUX_DTBXY_ generated by the L2OS processor, together with the most recent (ie current) AUX_DTBCUR, and outputs OTTs for use by future runs of the L2OS processor, and a new AUX_DTBCUR, used by the next run of the OTT post-processor.

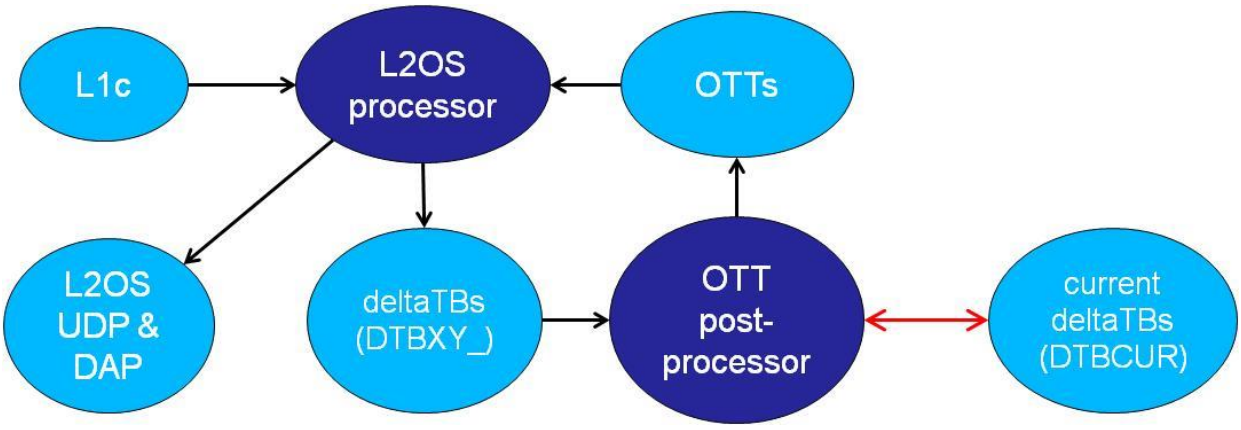

Figure 1: Overview of the SMOS L2 SSS processor and its files

An initially empty AUX_DTBCUR (in $SMOS_ROOT/products/AUX_/DTBCUR) is provided. Each time OSCOTT runs, if new OTT deltaTBs of sufficient quality are found in any of the input AUX_DTBXY_ products, they are merged with deltaTBs from AUX_DTBCUR, old ones discarded, and a new AUX_DTBCUR written, together with a set of new OTTs. For analysis, each AUX_DTBCUR also contains deltaTBs for the matching OTTs.

All rights reserved ARGANS © 2013

3 Installation

3.1 Installation of the OTT post-processor

The software is intended to be installed on a Linux 64 bit operating system environment. The Red Hat 5 Enterprise operating system is recommended because it is the environment on which the post-processor has been implemented and tested.

We assume hereafter that the operational processor, post-processor and all necessary auxiliary and input data files have already been installed and that an environment variable SMOS_ROOT has been set, pointing to the installation directory. Note that since the operational processor and post-processor are intended to be used together, it is assumed that they are both installed together with the same configuration and paths. To install, to test and to use the post-processor from source code, follow the procedure in the Operational Processor Software Release Note SO-RN-ARG-GS-0019.

3.2 Tree organization

After installation of the post-processor (& processor), the following file/directory structures will be accessible.

Type	Files	Path
Post-processor	Binary executable	*$SMOS_ROOT/devpp/OSCOTT_vv_vv.exe*
	Input configuration files, data and schema	*$SMOS_ROOT/smos/config/* *$SMOS_ROOT/smos/schemas/* *$SMOS_ROOT/smos/products/*
	Processor libraries	*$SMOS_ROOT/libpackages*

$SMOS_ROOT is an environment variable pointing to the post-processor (& processor) installation directory.

In order to run correctly, the post-processor needs the following hardware and software resources:

3.2.1 Hardware

- LINUX Operating System 64 bits (x86_64)
- 4 Gb of RAM
- At least 4Gb of disk space for installation (this does not include space needed for creating or running jobs)

3.2.2 Software

The SMOS L2 OS post-processor uses external libraries. They are included in the software package. Some libraries need licensing, such as the Earth Explorer CFI library. Therefore, you must accept the licenses of all third party software to use the SMOS L2 post- processor. Third party software are:

- Earth Explorer CFI software. Contact http://eop-cfi.esa.int/ to obtain the license.

All rights reserved ARGANS © 2013

- HDF5 library
- Log4c
- XML2, the XML C parser and toolkit, part of the Gnome project
- XML_RW_API from Indra Espacio (www.indra.es)
- GSL from Indra Espacio (www.indra.es)

All rights reserved ARGANS © 2013

4 L2 OS Post-Processor Dependency Model

The Dependency Model is the set of rules designed for each processor to generate its outputs based on specific combinations of input data files. This model tells the checking module which the valid relations are between files in order to successfully process SMOS data to upper processing levels.

The interaction between the L2 OS Post Processor and the DPGS is physically done through the interface between the PDPC-Core and the processing hosts where the post processor is installed. The PDPC-Core shall be responsible for providing the best set of inputs to the Post Processor in order it can process products with the best possible quality.

The PDPC-Core shall select the inputs accordingly to a configured set of rules for each executable. This set of selection rules is the dependency model. The orchestration of the execution of each post-processing tasks thus consists in the configuration of the PDPC-Core accordingly to the dependency model for the L2 OS Post Processor executable.

The number of inputs delivered for each product type should be the least possible to avoid increasing the transfer flow across the network.

Nominal post-processing will be launched as part of the SMOS processing chain by the PDPC-Core through its Processing Framework, providing the Post Processor with the required inputs. The list of inputs will be made available through the Job Order, which also gives the time periods in which each product should be used.

In the operational chain, the PDPC-Core will be responsible for running a single instance ot the L2 OS Post Processor job order with the associated input data at regular intervals (nominally once per day) in order to update the set of OTTs.

4.1 L2 OS Post Processor Auxiliary Data

L2 OS post-processing depends on the availability of the set of auxiliary data shown in the table below.

ID	Product description	Source	Update rate
AUX_DTBXY_	Delta TBs for each orbit intersecting regions of interest (ROIs) specified in AUX_CNFOSF	L2 OS processor	From each orbit intersecting ROIs
AUX_CNFOSF	Processor Configuration Parameters for Full Polarisation L2 Ocean Salinity	ARGANS	If required by ESLs
AUX_DTBCUR	Set of delta TBs used to make OTTs	OTT post-processor	Daily if configured in DPGS

Table 1: L2 OS Post Processor Auxiliary Data Files

All rights reserved ARGANS © 2013

For each auxiliary data file, the table shows the source and update rate. All the files listed are mandatory for processing.

4.2 Conventions

The following conventions are used in sections that follow when defining the dependencies:

- Mandatory/Optional: defines whether a supporting product is always needed for a processing task (mandatory) or the processing can still be performed in its absence through a backup solution (optional).

- Triggering: defines the conditions that must be met to launch the processing task.

4.3 OTT Post Processor processing

1. Executable to be called (by PDPC-Core): OSCOTT_<vv>_<rr>.exe

2. Processing Task:

- Processing a set of new and current deltaTB to a set of OTTs and new current deltaTBs as described in [R.D. 1] & [R.D. 3].

- To launch OTT post-processing the Job Order must contain:

    - one or more AUX_DTBXY_ data files
    - one AUX_DTBCUR data file
    - one AUX_CNFOSF data file

- Related Task_Table: deltaTB to OTT post-processing

3. Main Input Products: one or more AUX_DTBXY_ and one AUX_DTBCUR

4. Triggering: daily on ingestion of AUX_DTBXY_ in the PDPC-Core/PAS

5. List of supporting products needed:

□ AUX_DTBCUR: set of deltaTBs used to make OTTs
- Mandatory
- Dependency check:
    - Latest generation time as specified by the Fixed Header Creation_Date.

□ AUX_CNFOSF: Processor & Post-Processor Configuration Parameters for L2 Ocean Salinity
- Mandatory
- Dependency check:
    - Latest generation time as specified by the Fixed Header Creation_Date.

All rights reserved ARGANS © 2013

All rights reserved ARGANS © 2013

5 Operational configurations

5.1 Introduction

The SMOS L2 OS post-processor handles many files that are either input files or output files. Input files are loaded in the processor. They include either data or configuration information. A technical description of the content of all files handled by the post-processor is available in [R.D. 4] or [R.D. 11].

We focus hereafter on the configuration that is recommended for operational purpose (DPGS).

We detail below the main files:

- the task table file (CNF_TSKTBL)
- the private configuration file (CNF_OSCOTT)
- the main configuration file (AUX_CNFOSF) that contains:
  - Flag and Filter definitions
  - Regions of interest definitions
  - Post Processor configuration

5.2 The task table

The task table is:

SM_OPER_CNF_TSKTBL_20080801T000000_20500101T000000_vvv_rrr_1.xml

(vvv is the processor version, eg 610, and rrr is the release version, eg 001), and it can be found under the installation directory in:

$SMOS_ROOT/smos/task_tables/AUX_DTBXY_

5.3 Job orders

One job order is provided in the $SMOS_ROOT/Inputs directory to generate a set of OTT products from sample AUX_DTBXY_ and AUX_DTBCUR products:

SM_TEST_MPL_JOBORD_20050101T000000_20500101T000000_vvv_200_8.XML

5.4 The private configuration file

The private configuration is provided in $SMOS_ROOT/smos/products/CNF_/L2OS__:

SM_TEST_CNF_L2OS___20050101T000000_20500101T000000_vvv_rrr_8.EEF

All rights reserved ARGANS © 2013

5.5 The main configuration file (AUX_CNFOSF)

The main configuration file is shared between the L2 OS processor and OTT post-processor:

SM_TEST_AUX_CNFOSF_20050101T000000_20500101T000000_001_rrr_8.EEF

(rrr is the release version, eg 001), and it can be found under the installation directory in:

$SMOS_ROOT/smos/products/AUX_/CNFOSF

All rights reserved ARGANS © 2013

6 Return codes

When the post-processor stops, it returns with a specific code. The return code may help targeting the origin of errors. Below is a list of return codes and identification of the error source.

Two special return codes are 0 for normal termination of the processing and 255 when the processing is cancelled. Other return codes are grouped in two categories: if the return code is lower than 127, output products may be incomplete. Above 128, the return code indicates that no output products are generated but product reports (containing errors & warnings) are still produced.

Internal Error Name. L2 ouput product may be incomplete.	Return code	Function that generates the error and error description
ERROR_CODE_WRITE_OUTPUTS_WRITEFIXEDHEADER_SETELEMENTDATA	8,	writeFixedHeader.c, Cannot set data in fixed header
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_GETHEADER	9,	writeModule.c, Cannot get OTTxF/D header
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_EDITPRODUCT	10,	writeModule.c, Cannot edit product OTTxF/D
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_NODELETE	11,	writeModule.c, Cannot delete object
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_GETDATASET	12,	writeModule.c, Cannot get data set of OTTxF/D product
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_GETDATASETRECORD	13,	writeModule.c, Cannot get data set record of OTTxF/D product
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_WRITEDATASETRECORD	14,	writeModule.c, Cannot write data set record for OTTxF/D product
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_SETELEMENTDATA	15,	writeModule.c, Cannot set data for OTTxF/D product
ERROR_CODE_WRITE_OUTPUTS_WRITEOTTFILE_CREATEHEADER	16,	writeModule.c, Cannot create header for OTTxF/D product
ERROR_CODE_WRITE_OUTPUTS_WRITEDTBCURFILE_GETHEADER	17,	writeModule.c, Cannot get DTBCUR header
ERROR_CODE_WRITE_OUTPUTS_WRITEDTBCURFILE_CREATEHEADER	18,	writeMainHeader.c, Cannot get Node while writing main product header
ERROR_CODE_WRITE_OUTPUTS_WRITESPECIFICHEADER_GETNODE	19,	writeSpecificProductHeader.c, Cannot get node while writing SPH
ERROR_CODE_WRITE_OUTPUTS_WRITEMAINHEADER_GETNODE	20,	writeMainHeader.c, Cannot get Node while writing main product header
ERROR_CODE_WRITE_OUTPUTS_WRITEFIXEDHEADER_GETNODE	21,	writeFixedHeader.c, Cannot get Node while writing fixed header
ERROR_CODE_WRITE_OUTPUTS_WRITESPECIFICHEADER_SETOUTERDIM	22,	writeSpecificProductHeader.c, Cannot set outer dimension
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_CREATEPRODUCT	23,	writeReport.c, Cannot create product report

All rights reserved ARGANS © 2013

Internal Error Name. L2 ouput product may be incomplete.	Return code	Function that generates the error and error description
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_CREATEHEADER	24,	writeReport.c, Cannot create product header
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_CREATEDATABLOCK	25,	writeReport.c, Cannot create data block
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_GETNODE	26,	writeReport.c, Cannot get Node while writing product report
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_SETOUTERDIM	27,	writeReport.c, Cannot set outer dimension
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_WRITEHEADER	28,	writeReport.c, Cannot write header
ERROR_CODE_WRITE_OUTPUTS_WRITEREPORT_WRITEDATABLOCK	29,	writeReport.c, Cannot write data block
ERROR_CODE_WRITE_OUTPUTS_SET_ELEMENT_DATA	30,	setElementData.c, Cannot set element data

Internal Error Name of return code. No L2 product generation.	Exit code	Function that generates the error and possible error description
ERROR_CODE_MAIN_WRONG_ARG	128,	main.c, wrong number of arguments passed to the executable
ERROR_CODE_TOOLS_CFIDATEFUNCTIONS_PROCESSINGFROMTRANSPORTTIME	132,	CFIDateFunctions.c/getProcessingTimeFromTransportTime, SS CFI function xl_time_transport_to_processing returns in error
ERROR_CODE_TOOLS_CFIDATEFUNCTIONS	132,	CFIDateFunctions.c, Error occured in CFIDateFunctions.c file
ERROR_CODE_CONFIG_PROTO_READITERCOEF_RETRIEVPARAMID	134,	readIterCoef.c, readIterCoef function fails to read retrievParamId
ERROR_CODE_CONFIG_PROTO_READITERCOEF_NOOPEN	134,	readIterCoef.c, Cannot open the product CNFOS2
ERROR_CODE_CONFIG_PROTO_READITERCOEF_NODATA	134,	readIterCoef.c, Cannot load IterCoef data
ERROR_CODE_CONFIG_PROTO_READITERCOEF_ALLOC_LUT	134,	readIterCoef.c, fail to allocate memory for Itercoef lut
ERROR_CODE_CONFIG_PROTO_READITERCOEF_GET_NODE	134,	readIterCoef.c, Cannot get node
ERROR_CODE_CONFIG_PROTO_READITERCOEF_GETNODEDATA	134,	readIterCoef.c, Cannot get node data.
ERROR_CODE_TOOLS_TOOLS_INVERSEDIAGMAT_NONSQUAREMAT	137,	tools.c, inverseDiagMat function, dimensions of input matrix are not equal
ERROR_CODE_TOOLS_TOOLS_INVERSEDIAGMAT_ALLOCMAT	138,	tools.c, inverseDiagMat function, fail to allocate memory for matrix invDiag
ERROR_CODE_TOOLS_TOOLS_INVERSEDIAGMAT_NONDIAGMAT	138,	tools.c, inverseDiagMat function, input matrix is not diagonal. Found non-diagonal term different from 0
ERROR_CODE_TOOLS_TOOLS_TRANSPOSEMAT_ALLOCMAT	139,	tools.c, transposeMat function, fail to allocate memory for matrix st

All rights reserved ARGANS © 2013

Internal Error Name of return code. No L2 product generation.	Exit code	Function that generates the error and possible error description
ERROR_CODE_TOOLS_TOOLS_SUMMAT_SIZEMATS	140,	tools.c, sumMat function, inputs matrices are not with the same dimensions
ERROR_CODE_TOOLS_TOOLS_SUMMAT_ALLOCMAT	140,	tools.c, sumMat function, fail to allocate memory for matrix sum
ERROR_CODE_TOOLS_TOOLS_DIFFMAT_SIZEMATS	141,	tools.c, diffMat function, inputs matrices are not with the same dimensions
ERROR_CODE_TOOLS_TOOLS_DIFFMAT_ALLOCMAT	141,	tools.c, diffMat function, fail to allocate memory for matrix diff
ERROR_CODE_TOOLS_TOOLS_PRODMAT_SIZEMATS	142,	tools.c, prodMat function, input matrices are not with compatible dimensions
ERROR_CODE_TOOLS_TOOLS_PRODMAT_ALLOCMAT	142,	tools.c, prodMat function, fail to allocate memory for matrix prod
ERROR_CODE_TOOLS_TOOLS_SCALPRODMAT_ALLOCMAT	143,	tools.c, scalProdMat function, fail to allocate memory for matrix scalProd
ERROR_CODE_TOOLS_TOOLS_KEEPDIAGMAT_SIZEMAT	144,	tools.c, keepDiagMat function, dimensions of input matrix are not equal
ERROR_CODE_TOOLS_TOOLS_KEEPDIAGMAT_ALLOCMAT	144,	tools.c, keepDiagMat function, fail to allocate memory for matrix diag
ERROR_CODE_TOOLS_TOOLS_INVERSEMAT_SIZEMAT	145,	tools.c, inverseMat function, dimensions of input matrix are not equal
ERROR_CODE_TOOLS_TOOLS_INVERSEMAT_ALLOCMAT_LL	145,	tools.c, inverseMat function, fail to allocate memory for matrix ll
ERROR_CODE_TOOLS_TOOLS_INVERSEMAT_ALLOCMAT_ILL	145,	tools.c, inverseMat function, fail to allocate memory for matrix ill
ERROR_CODE_TOOLS_TOOLS_INVERSEMAT_LLNOTDEFPOSITIVE	145,	tools.c, inverseMat function, ll matrix is not definite positive
ERROR_CODE_TOOLS_INTERPHERMIT_NSAMPLINGTOOLOW	146,	interpHermit.c, not enough sampling in the dimension for Hermit interpolation
ERROR_CODE_TOOLS_INTERPHERMIT_INDEXINVALID	146,	interpHermit.c, index for interpolation is not valid
ERROR_CODE_CONFIG_PROTO_READPARAMINDICES_NOOPEN	152,	readParamIndices.c, Cannot get product informations of AUX_CNFOSD/F
ERROR_CODE_CONFIG_PROTO_READPARAMINDICES_NONODE	152,	readParamIndices.c, Cannot get node
ERROR_CODE_CONFIG_PROTO_READPARAMINDICES_NODATA	152,	readParamIndices.c, Cannot load ParamIndices data
ERROR_CODE_CONFIG_PROTO_READPARAMINDICES_MULTIINDEX	152,	readParamIndices.c, Parameter index not unique
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READHEADER_NONODE DATA	159,	readHeader.c, Cannot get data from node
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READHEADER_NONODE	159,	readHeader.c, Cannot get node
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READDTBCUR_NOOPEN	160,	readDTBCUR.c, Cannot read AUX_DTBCUR

All rights reserved ARGANS © 2013

Internal Error Name of return code. No L2 product generation.	Exit code	Function that generates the error and possible error description
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READDTBCUR_NODATA	161,	readDTBCUR.c, Cannot read AUX_DTBCUR
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READDTBCUR_NOCOUNT	162,	readDTBCUR.c, Cannot get the count of data blocks
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READDTBXY_NOOPEN	163,	readDTBXY.c, Cannot read AUX_DTBXY
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READDTBXY_NODATA	164,	readDTBXY.c, Cannot read AUX_DTBXY
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READDTBXY_NOCOUNT	165,	readDTBXY.c, Cannot get the count of data blocks
ERROR_CODE_CONFIG_PROTO_READJOBORDER_NODATA	167,	readJobORder.c, Cannot find data in the job order file
ERROR_CODE_CONFIG_PROTO_READJOBORDER_NONODE	167,	readJobORder.c, Cannot find tag in the job order file, cannot find node
ERROR_CODE_CONFIG_PROTO_READJOBORDER_VARNOTSET	167,	readJobORder.c, Cannot find tag in the job order file, cannot set variable
ERROR_CODE_CONFIG_PROTO_READJOBORDER_NOOPEN	167,	readJobORder.c, Cannot open job order file
ERROR_CODE_CONFIG_PROTO_NOMEM	168,	config_proto.c, Cannot allocate memory
ERROR_CODE_CONFIG_PROTO_READCNF_L2OS_NOCLOSE	169,	readCNF_L2O.c, Cannot close private configuration file CNF_L2OS
ERROR_CODE_CONFIG_PROTO_READCNF_L2OS_NODATA	169,	readCNF_L2O.c, Cannot retrieve data from file CNF_L2OS
ERROR_CODE_CONFIG_PROTO_READCNF_L2OS_NOOPEN	169,	readCNF_L2O.c, Cannot open private configuration file CNF_L2OS
ERROR_CODE_CONFIG_PROTO_READCNF_L2OS_XML_RW_API_HOME_ENVNOTFOUND	169,	readCNF_L2O.c, environment vaiable XML_RW_API_HOME not set. Unable to find private configuration file CNF_L2OS
ERROR_CODE_CONFIG_PROTO_READCNF_L2OS_SMOS_CNF_L2OS_ENVNOTFOUND	169,	readCNF_L2O.c, environment vaiable SMOS_CNF_L2OS not set. Unable to find private configuration file CNF_L2OS
ERROR_CODE_READ_INPUTS_DATA_SELECTION_GET_VALIDITY_PERIOD	170,	get_validity_period.c, fail to get validity period
ERROR_CODE_CHECK_VALIDITY_PERIOD_PRODUCT_NOTVALID	171,	check_validity_period.c, invalidity period of the product.
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READLUT	172,	readLut.c, Unable to read a Lut.
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READLUT_ALLOCLUT	173,	readLut.c, fail to allocate memory for a LUT
ERROR_CODE_READ_INPUTS_DATA_SELECTION_GETPRODUCT_INFO	173,	readXmlFunctions.c, fail to get product informations
ERROR_CODE_FORWARD_MODEL_SUPPORT_RADEC	175,	RADec.c, fail to compute right ascension and declination
ERROR_CODE_DIVISION_BY_ZERO	176,	mathematical exception

All rights reserved ARGANS © 2013

Internal Error Name of return code. No L2 product generation.	Exit code	Function that generates the error and possible error description
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READTHRESHOLDS_NOOPEN	180,	readThresholds.c, Cannot get information of product CNFLOS2
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READTHRESHOLDS_NONODE	180,	readThresholds.c, Cannot get node
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READTHRESHOLDS_NODATA	180,	readThresholds.c, Cannot load data for THRESHOLDS
ERROR_CODE_GENERIC_INTERP_INEXISTANT_ARRAY_DIMENSION	183,	interp.c, Error occured in GenericInterp function: inexistent array dimension
ERROR_CODE_GENERIC_INTERP_INVALID_INDEX	184,	interp.c, Error occured in GenericInterp function: invalid index
ERROR_CODE_GENERIC_INTERP_INVALID_COEFFICIENT	185,	interp.c, Error occured in GenericInterp function: invalid coefficient
ERROR_CODE_GENERIC_INTERP_VECT_INVALID_DIMENSION	186,	interp.c, Error occured in GenericInterpVect function: invalid dimension
ERROR_CODE_NEAREST_NEIGHBOUR_OUT_OF_MEMORY	187,	tools.c, Error occured in nearestNeighbour function: out of memory
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READFLAGS_NODATA	195,	filters.c, Cannot get flags data from AUX_CNFOSF/D
ERROR_CODE_READ_INPUTS_DATA_SELECTION_READFILTERS_BADDATA	195,	filters.c, Cannot get filter data from AUX_CNFOSF/D
ERROR_CODE_TOO_MANY_MESSAGES	250,	Number of messages > MAX_MESSAGES
ERROR_CODE_SIGILL	251,	trapped SIGILL - illegal instruction
ERROR_CODE_SIGBUS	252,	trapped SIGBUS - bad memory reference
ERROR_CODE_SIGSEGV	253,	trapped SIGSEGV - invalid memory reference
ERROR_CODE_SIGFPE	254,	trapped SIGFPE - floating point exception (including div 0 & overflow)
RETURN_CODE_CANCEL_ORDER	255	Order cancelled (via Ctrl-C or cancel.xml)

All rights reserved ARGANS © 2013

7 GSL Log INFO Messages

The SMOS L2 OS post-processor outputs several types of log messages: advance, info, debug, warning & error. The default setting for the Log_Level specified in the job order is INFO; at this level it is required that a limit of approximately 10 INFO log messages are sent to the GSL (in addition to the advance messages, and any error or warning messages). During normal processing, exactly 2 ADVANCE messages are generated: START & END. ERROR messages result in processor termination. Any number of WARNING messages may be issued during processing.

The following INFO messages are sent to the GSL in this sequence if there are no processing errors:

INFO message	**Notes**
Start of processing	Immediately after [A][000_STA_000]
Reading DTBCUR file...	
Reading %d DTBXY files...	
Start of main processing with deltaTBs from %d ascending and %d descending orbits	
Writing output files...	
End of processing	Followed by [A][100_END_000]

All rights reserved ARGANS © 2013